

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mark Hočevar

**Termostat s funkcijami za nadzor na
daljavo**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Uroš Lotrič

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Na področju bivanja se vedno bolj uveljavljajo avtomatizirane rešitve z namenom povečanja udobja in izboljšanja kakovosti življenja. Želja mnogih uporabnikov je uporabniku prijazna rešitev za daljinski nadzor ogrevanja objekta. Razvijte prototip splošno namenske naprave, pri čemer se osredotočite na cenovno ugodno rešitev z dobro uporabniško izkušnjo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Mark Hočevar, z vpisno številko **63110199**, sem avtor diplomskega dela z naslovom:

Termostat s funkcijami za nadzor na daljavo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Uroša Lotriča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 10. septembra 2014

Podpis avtorja:

*Zahvaljujem se mentorju izr. prof. dr. Urošu Lotriču za pomoč pri izdelavi
diplomskega dela. Zahvaljujem se tudi moji družini, ki me vedno podpira in
spodbuja.*

Kazalo

Slike

Povzetek

Abstract

1	Uvod	1
2	Termostat	3
3	Uporabljena strojna oprema	7
3.1	Raspberry Pi	7
3.2	Rele	9
3.3	Termometer za priklop na vodilo 1-Wire	10
3.4	Modem GSM	12
3.5	Testno vezje	12
4	Programski del	15
4.1	Aplikacija na računalniku Raspberry Pi	15
4.1.1	Komunikacija	16
4.1.2	Termostat	17
4.1.3	Periodično beleženje	19
4.2	Mobilna aplikacija	22
4.2.1	Stanje	23
4.2.2	Nastavitve	25

KAZALO

4.2.3	Pomoč	27
5	Rezultati	29
6	Zaključek	33

Slike

2.1	Histerezna zanka. Stanje nič pomeni izklop stanje ena pa vklop.	4
2.2	Termostat mars 2	5
2.3	Termostat BPT322-GST	5
3.1	Raspberry Pi model B verzija 1	9
3.2	Rele za kontrolo centralnega ogrevanja	10
3.3	Shema parazitne vezave termometra	11
3.4	Shema normalne vezave termometra	11
3.5	Termometer za priklop na vodilo 1-Wire DS18B20	11
3.6	Huawei E220 usb modem	12
3.7	Ženska letvica	13
3.8	Testno vezje	13
3.9	Shema testnega vezja	14
4.1	Prikaz beleženja dogodkov	19
4.2	Prikaz beleženja napak	20
4.3	Prikaz beleženja sporočil SMS	21
4.4	Prikaz beleženja temperature	21
4.5	Prikaz postavitve zavihka stanje	23
4.6	Prikaz okenca za spremembo temperature	24
4.7	Prikaz spremenjenega elementa	25
4.8	Prikaz postavitve zavihka nastavitve	26
4.9	Prikaz postavitve zavihka pomoč	27

SLIKE

5.1	Grelno telo za testiranje	29
5.2	Prikaz testnega vezja med delovanjem	31
5.3	Prikaz aplikacije med delovanjem	31

Povzetek

Diplomsko delo govori o izdelavi termostata. Gre za termostat z možnostjo oddaljenega upravljanja. Osnova termostata je majhen računalnik Raspberry Pi. Za merjenje temperature uporabljamo digitalni termometer, za preklapljanje ogrevanja pa uporabljamo rele. Da se izognemo prekomernemu preklapljanju releja, smo uporabili histerežno zanko. Termostat nam omogoča tudi upravljanje digitalnih vhodov in izhodov.

Oddaljeno upravljanje je realizirano s pomočjo sporočil SMS. Za lažje upravljanje termostata smo razvili aplikacijo za operacijski sistem Android. Aplikacija nam omogoča uporabo grafičnega umesnika za izdelavo sporočil SMS, s katerimi nadzorujemo termostat.

Ključne besede: Raspberry Pi, termostat, histerezna zanka, rele, termometer za priklop na vodilo 1-Wire, aplikacija Android.

Abstract

This bachelor Thesis talks about developing a thermostat. It is a thermostat with possibility of remote control. The base of thermostat is credit card sized computer Raspberry Pi. We use digital thermometer for temperature measurement and relay for heating control. To avoid over switching of relay we use a hysteresis loop. We can also control digital input and outputs.

For remote control of thermostat we use SMS messages. For easier control we have developed Android application, which allows us to use graphical interface for creating SMS messages in order to control the thermostat.

Keywords: Raspberry Pi, thermostat, hysteresis loop, relay, 1-Wire digital thermometer, Android application.

Poglavje 1

Uvod

Ljudje smo navajeni udobja. Ena izmed stvari, ki jih v današnjem času jemljemo kot samoumevno, so prijetno ogrevana stanovanja. To predvsem pride do izraza pozimi, ko so temperature zunaj nizke. Pomemben del sodobnega ogrevalnega sistema je termostat. V okviru diplomske naloge smo razvili termostat z dodatnimi funkcionalnostmi, med katerimi je glavna funkcija oddaljen nadzor preko sporočil SMS. Razlog za željo po termostatu, ki ga lahko upravljamo oddaljeno, je enostaven in sicer udobje. Poglejmo primer, v katerem je oddaljeno upravljanje termostata zelo uporabno. Imamo vikend v gorah. V zimskih mesecih se zunanja temperatura spusti pod 0 °C. Na vikendu je centralno ogrevanje nastavljeno na nekje 5 °C, zato da nam voda v ceveh ne zmrzne. Ko se v petek po službi odpravimo na vikend, tja pridemo v mrzlih 5 °C. Temu se lahko izognemo z daljinskim vklopom ogrevanja. V petek dopoldne oziroma, ko se odločimo, da bomo šli na vikend, preko mobilnega telefona nastavimo termostat na temperaturo 20 °C in ob našem prihodu je na vikendu lepo toplo. Naš končni izdelek omogoča prav to, s pomočjo mobilne aplikacije lahko oddaljeno nadziramo termostat. Ima pa naš termostat tudi dodano možnost nastavljanja in kontroliranja digitalnih vhodov in izhodov.

V prvem delu diplome bomo opisali osnovno delovanje termostata in nekaj že obstoječih rešitev. Drugi del je namenjen predstavitvi strojne opreme,

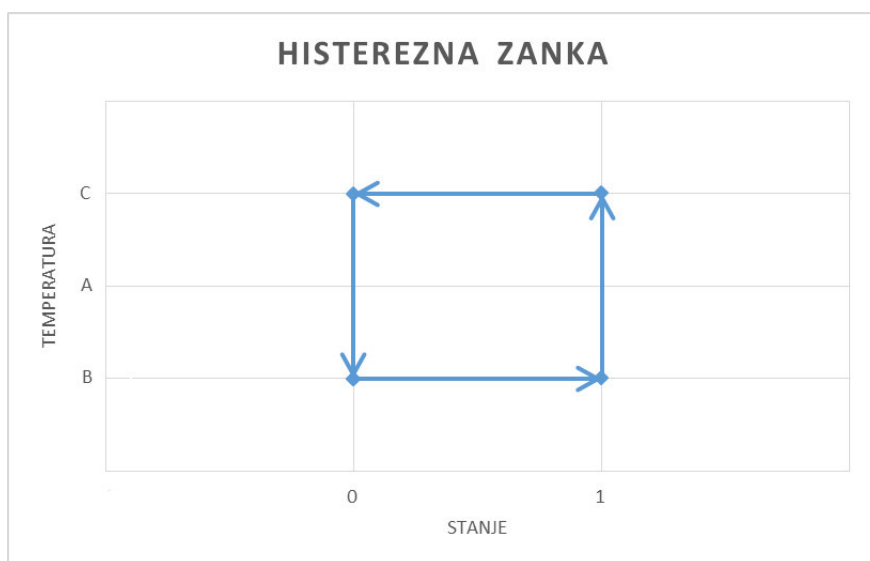
ki smo jo uporabili za izdelavo projekta. Sledi opis programskega dela ter rezultati testiranja. Na koncu pride še zaključek, v katerem bomo pogledali, kaj bi lahko pri projektu spremenili oziroma dodali.

Poglavje 2

Termostat

Termostat je del ogrevalnega sistema. Skrbi za vzdrževanje temperature na želeni vrednosti. Termostat izmeri temperaturo in s pomočjo prižiganja ter ugašanja grelca posredno kontrolira temperaturo. Če bi za pogoj ogrevanja gledali samo trenutno in ciljno temperaturo, bi prišlo do prekomernega preklapljanja, kar bi zelo skrajšalo življenjsko dobo releja, ki skrbi za preklapljanje. Torej da do tega ne pride se uporablja histerezna zanka [5, 12].

Histerezna zanka za izračun izhoda ne uporablja samo trenutnega stanja, ampak tudi prejšnje stanje. Gre za enostaven algoritem, ki ga prikazuje slika 2.1. Če hočemo imeti temperaturo A , se bo ogrevanje vključilo šele pri temperaturi B , ki je manjša od temperature A . Ogrevanje pa se bo izključilo šele pri temperaturi C , ki je večja od A [7].



Slika 2.1: Histerezna zanka. Stanje nič pomeni izklop stanje ena pa vklop.

Enostavnih termostatov, ki omogočajo nastavljanje ciljne temperature, je veliko. Taki termostati tudi niso dragi, najbolj osnovne verzije dobimo že za 20 evrov. Vse se zaplete, ko pogledamo ponudbo termostatov, ki omogočajo oddaljeni nadzor. Večina teh ima oddaljen nadzor realiziran preko internetne povezave, naša želja pa je nadzor preko sporočil SMS. Prav tako je naša želja možnost nadzora digitalnih vhodov in izhodov preko termostata. Našli smo tudi nekaj podobnih termostatov, vendar noben ni popolnoma ustrezal našim željam.

Prvi tak termostat je termostat Mars 2, ki ga izdeluje Siop Elektronika. Paket vsebuje dve napravi, in sicer sobni termostat z vgrajenim modulom GSM in sprejemnik, ki nadzoruje ogrevanje. Povezava med moduloma je brezžična. Termostat podpira upravljanje preko sporočil SMS, ne podpira pa nadzora digitalnih vhodov in izhodov. Cena termostata Mars 2 v času pisanja znaša 410 evrov [3].



Slika 2.2: Termostat mars 2

Naslednji primeren termostat je termostat BPT322-GST. To je napreden termostat z veliko funkcijami. V paketu dobimo termostat in preklopno enoto. Oba elementa med seboj brezžično komunicirata. Ima možnost nadzora preko sporočil SMS. Omogoča tudi programiranje temperature po dnevih in urah. Enota ima zaslon LCD, ki je osvetljen za lažje branje. Tudi ta termostat ne podpira digitalnih vhodov in izhod. Cena v času pisanja znaša 240 evrov [2].



Slika 2.3: Termostat BPT322-GST

Pri obstoječih rešitvah nas tako moti njihova visoka cena in pa odsotnost možnosti nadziranja digitalnih vhodov in izhodov. Digitalne vhode lahko uporabimo s senzorjem za vrata in s tem preverimo, če so katera izmed vrat odprta. Na digitalne izhode pa lahko povežemo luči. Noben od izdelkov tudi nima mobilne aplikacije za nadzor, s katero si poenostavimo oddaljeno upravljanje.

Poglavje 3

Uporabljena strojna oprema

Pri izdelavi pametnega termostata smo uporabili spodaj opisano strojno opremo. Osnova termostata je majhen računalnik Raspberry Pi. Preko priključka USB je nanj priklopljen modem USB Huawei E220, ki skrbi za komunikacijo. Preko pinov GPIO (general purpose input output pins - Splošno namenski vhodno izhodni pini) so priklopljene ostale dodatne naprave kot so termometer, rele, diode LED in stikala. Pini GPIO so pini, ki jih lahko programsko nadziramo. Vsak pin lahko poljubno določimo kot vhod ali izhod.

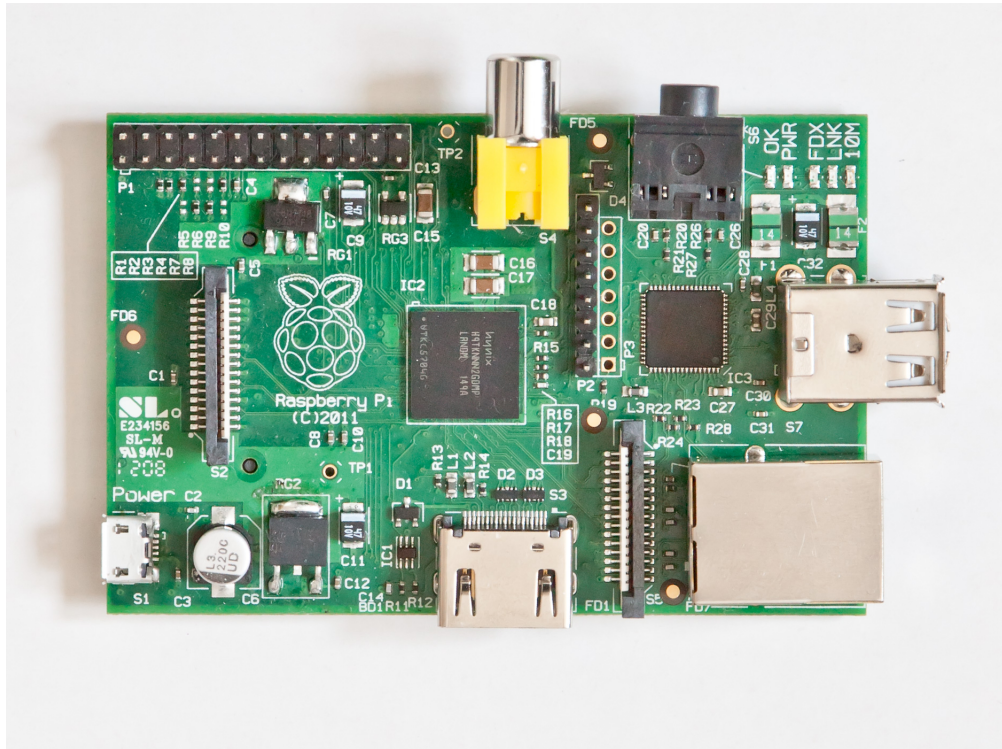
3.1 Raspberry Pi

Raspberry Pi je majhen računalnik, ki je bil razvit za namen izobraževanja. Gre za tako imenovani kartični računalnik, saj ni veliko večji od bančne kartice. Osnova računalnika Raspberry Pi je sistem na čipu Broadcom BCM2835. Sistem na čipu vsebuje vse osnovne elemente računalnika na enem samem čipu. Prednost takega sistema je nizek strošek izdelave [4]. Omenjen čip vsebuje procesorsko jedro ARM, grafično jedro VideoCore IV in 256 oziroma 512 MB pomnilnika RAM. Obstaja več različnih modelov računalnika Raspberry Pi, ki se ne razlikujejo veliko. Vsi modeli imajo izhod HDMI, kompozitni izhod RCA, 3,5 mm zvočni izhod in pine GPIO. Model A je najbolj oskubljen model. Ima 256 MB pomnilnika RAM, samo en priključek USB in nima

priključka Ethernet. Model B ima dve verziji, ki se razlikujeta po velikosti pomnilnika RAM. Verzija 1 ima 256 MB, verzija 2 pa 512 MB pomnilnika RAM. Model B ima dva priključka USB in tudi priključek Ethernet [1]. Obstaja še prenovljena verzija modela B, ki se imenuje B+. Model B+ ima kar štiri priključke USB, lepše razporejene komponente na vezju, tako da so razni priključki (USB, Ethernet itd.) lepše poravnani. Za projekt smo uporabili model B verzije 1.

Na računalniku teče prirejen operacijski sistem Linux Raspbian, ki se zažene iz kartice SD. Raspbian je distribucija Linuxa, katere osnova je distribucija Debian in je optimizirana za poganjanje na računalniku Raspberry Pi.

Za računalnik Raspberry Pi smo se odločili zaradi več razlogov. Glavni razlog je predvsem to, da na njem lahko poganjamo operacijski sistem Linux. To nam omogoča enostavno uporabo modema GSM na priključku USB. Lahko bi uporabljali, kakšno izmed razvojnih ploščic ARM, na primer STM32F4 Discovery, vendar v tem primeru ne bi mogli uporabiti modema USB. Drugi razlog je cena. Raspberry Pi stane okoli 36 evrov, kar je za tako majhen in zmogljiv računalnik zelo malo. Če primerjamo ceno podobnega majhnega računalnika, ki tudi omogoča poganjanje operacijskega sistema Linux, BeagleBone Black pri okoli 50 evrov, vidimo, da je Raspberry Pi ugodnejši. Raspberry Pi ima tudi pine GPIO, ki jih potrebujemo za priklop digitalnega termometra, releja in digitalnih vhodov ter izhodov. Dobra lastnost je tudi majhna poraba električne energije, saj za svoje delovanje potrebuje le nekje med 1.5-3 W.



Slika 3.1: Raspberry Pi model B verzija 1

Vir: [8].

3.2 Rele

Glavna funkcija termostata je prižiganje in ugašanje centralnega ogrevanja. Zaradi premajhne izhodne moči pinov GPIO, centralnega ogrevanje ne moremo prižigati neposredno, zato za prekllop uporabimo rele. Rele je elektromagnetno stikalo, ki ga vklapljam in izklapljam s krmilno napetostjo. Na ta način lahko s krmilnim signalom nizke napetosti (v našem primeru 3.3 V) vklapljam in izklapljam velike porabnike. Za prižiganje in ugašanje centralnega ogrevanja smo uporabili rele SONGLE SRD-05VDC-SL-C. Rele omogoča preklapljanje do 10 A in 30 V enosmerne napetosti ter do 10 A in 250 V izmenične napetosti.



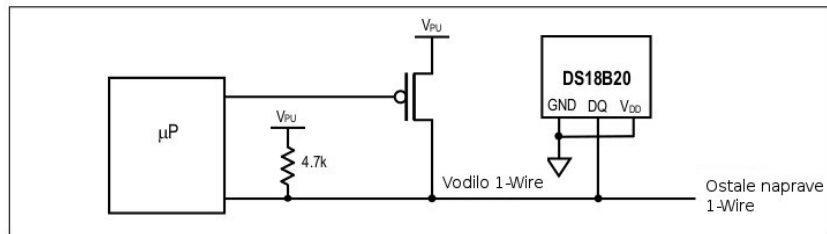
Slika 3.2: Rele za kontrolo centralnega ogrevanja

Vir: [9].

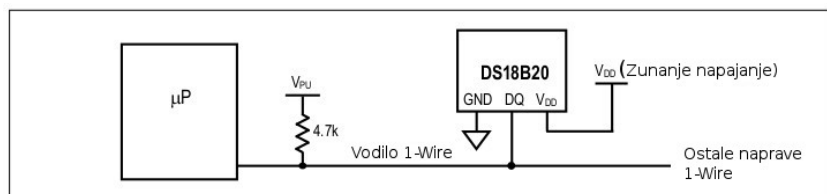
3.3 Termometer za priklop na vodilo 1-Wire

Za merjenje temperature smo uporabili termometer s protokolom 1-Wire. Ker Raspberry Pi nima vgrajenega analogno - digitalnega pretvornika, ne moremo uporabiti termo upora, ki se ponavadi uporablja za merjenje temperature. Problem bi lahko rešili z dodatnim vezjem za analogno - digitalno pretvarjanje. Zaradi enostavnosti smo se raje odločili za uporabo termometra s protokolom 1-Wire. Protokol 1-Wire omogoča digitalno komunikacijo po eni žici, od tod tudi ime. Termometer ima tri pine, pin za napajanje, pin za komunikacijo in pin za ozemljitev. Imamo dve možnosti povezave. Termometer lahko povežemo samo na pin za ozemljitev in pin za komunikacijo. Temu rečemo tudi parazitna vezava. V tem primeru se termometer napaja preko istega pina, preko katerega poteka komunikacija. Shemo parazitne vezave vidimo na sliki 3.3. Drugi način vezave pa uporablja tudi pin za napajanje. V tej vezavi lahko dosežemo daljše razdalje komunikacije in boljše delovanje termometra, zato smo uporabili to vezavo. Shemo take vezave vidimo na sliki 3.4. Uporabili smo termometer proizvajalca Maxim in sicer model DS18B20. Razpon termometra je od -55°C do $+125^{\circ}\text{C}$. Njegova natančnost je $0,5^{\circ}\text{C}$ na območju od -10°C do $+85^{\circ}\text{C}$. Ker bomo merili temperaturo

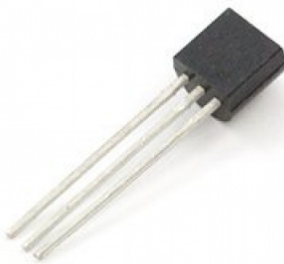
v prostoru, nam razpon in natančnost termometra povsem zadostujeta [6].



Slika 3.3: Shema parazitne vezave termometra



Slika 3.4: Shema normalne vezave termometra



Slika 3.5: Termometer za priklop na vodilo 1-Wire DS18B20

Vir: [10].

3.4 Modem GSM

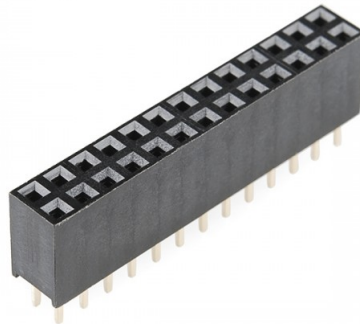
Vezje bomo oddaljeno upravljali, preko sporočil SMS, za kar potrebujemo modem GSM. Uporabili smo Huawei E220 modem, ki je na Raspberry Pi povezan preko vmesnika USB. Modem E220 smo izbrali predvsem zaradi njegove razširjenosti in podprtosti v Linux sistemih in seveda podpore prejetanja in pošiljanja sporočil SMS. Za povezavo z modemom smo uporabili knjižnico libgammu. Knjižnica vsebuje funkcije, s katerimi se lahko povežemo in upravljamo modem.



Slika 3.6: Huawei E220 usb modem

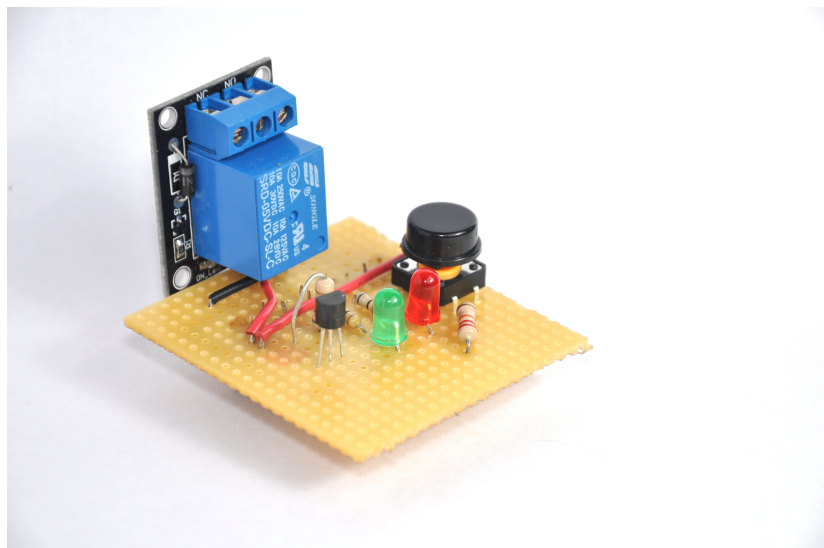
3.5 Testno vezje

Za namene testiranja smo izdelali testno vezje. Vezje je sestavljeno iz ženske letvice dolžine trinajst in širine dve, ki jo vidimo na sliki 3.7.

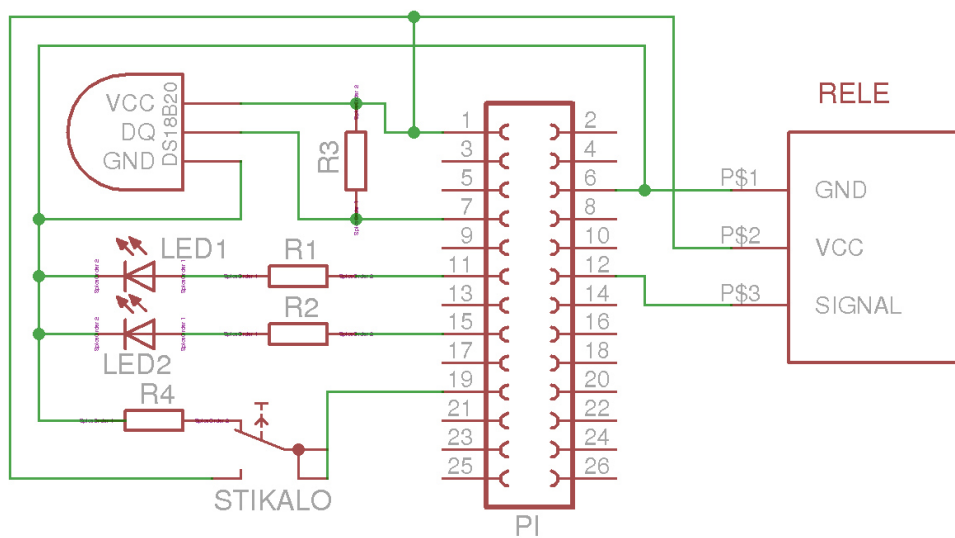


Slika 3.7: Ženska letvica

Letvica služi za priklop na pine GPIO na računalniku Raspberry Pi. Na vezju imamo termometer s protokolom 1-Wire. Signalni pin termometra je povezan na pin sedem na letvici. Na vezju imamo rdečo in zeleno diodo LED. Diodi sta preko upora povezani na pina enajst in petnajst. Diodi služita za preverjanje delovanja digitalnih izhodov. Digitalni vhod preverjamo z gumbom, ki je povezan na pin devetnajst. Na vezju imamo tudi rele, ki ima signalni pin povezan na pin osem. Vezje vidimo na sliki 3.8, shemo vezja pa na sliki 3.9.



Slika 3.8: Testno vezje



Slika 3.9: Shema testnega vezja

Poglavje 4

Programski del

Celoten projekt lahko razdelimo na dva večja dela. Prvi del teče na računalniku Raspberry Pi. Na njem tečeta dva ločena programa. Oba programa sta tako imenovana demoni. Demon je računalniški program, ki teče v ozadju. Uporaba demoni je smiselna, ker ne potrebujemo direktnega nadzora iz strani uporabnika [11]. Prvi program skrbi za prejemanje in pošiljanje sporočil SMS, drugi program pa za nadzor termostata.

Drugi del je mobilna aplikacija za operacijski sistem Android. Android je operacijski sistem za mobilne telefone. Aplikacija nam omogoča, da s pomočjo telefona z operacijskim sistemom Android nadziramo termostat. V tej aplikaciji lahko nastavljamo različne parametre termostata, kot na primer ciljno temperaturo.

4.1 Aplikacija na računalniku Raspberry Pi

Na računalniku Raspberry Pi tečeta dva demoni. Prvi skrbi za komunikacijo preko sporočil SMS, drugi pa za delovanje termostata. Komunikacija med obema programoma se izvaja s pomočjo tekstovnih datotek in Linux signalov. Linux signali se uporabljajo za medsebojno komunikacijo programov oziroma procesov. Obstaja več različnih signalov, ki imajo v naprej definirane uporabe. V programu za komunikacijo uporabljamo signala SIGUSR1

in SIGUSR2. To sta signala, ki sta definirana kot uporabniško definirana signala. To pomeni, da njuna akcija ni v naprej določena. Signal SIGUSR1 uporabljamo kot signal, ki sporoči drugemu programu, naj prebere datoteko in naredi kar je v njej zapisano. Signal SIGUSR2 uporabljamo za potrjevanje, da smo datoteko prebrali. Vsak od programov uporablja svojo datoteko, v katero piše ukaze oziroma sporočila drugemu programu. Torej, ko program zapiše nek ukaz v svojo datoteko, to drugemu programu sporoči preko signala SIGUSR1. Ko drug program datoteko prebere, mu to potrdi s signalom SIGUSR2. Potrjevanje s signali smo realizirali, da ne pride do istočasne uporabe iste datoteke za sporočanje. Tako smo preprečili, da se kakšen ukaz ne izbriše.

Za dva programa smo se odločili zaradi modularnosti. S tem smo ločili del, ki skrbi za komunikacijo, in del, ki skrbi za delovanje termostata. Če se odločimo, da bi namesto sporočil SMS uporabljali internetno povezavo, enostavno zamenjamo samo program za komunikacijo. Oba programa imata tudi beleženje pomembnih dogodkov za lažji nadzor in pregled dogajanja.

4.1.1 Komunikacija

Prvi program na računalniku Raspberry Pi skrbi za komunikacijo. To zajema predvsem sprejemanje sporočil SMS, njihovo dekodiranje ter pošiljanje odgovorov v obliki sporočil SMS.

Za upravljanje modema uporabljamo knjižnico libgammu. Libgammu je odprto kodna knjižnica, ki omogoča komunikacijo z določenimi telefoni in modemi. S pomočjo libgammu knjižnice vključimo modem USB, preverimo, ali smo dobili kakšno novo sporočilo SMS, in pošiljamo sporočila SMS. Kljub temu, da knjižnica libgammu omogoča ogromno drugih funkcionalnosti, uporabljamo samo funkcionalnosti za upravljanje s sporočili SMS.

Osnovna naloga programa je periodično preverjanje, ali je modem prejel kakšno novo sporočilo SMS. Ko program prejme novo sporočilo SMS, najprej preveri telefonsko številko. Zaradi varnosti je v programu realizirana baza telefonskih števil, iz katerih lahko sprejema ukaze. S tem onemogočimo

možnost, da bi nekdo iz nepotrjene številke upravljal s termostatom. Naslednji korak je preverjanje, če sporočilo SMS vsebuje ključno besedo [termostat]. Vsako sporočilo SMS, ki je namenjeno termostatu, se mora začeti s ključno besedo [termostat]. S tem se izognemo potencialnim problemom, če termostat dobi sporočilo SMS, ki ni namenjeno njemu. Po preverjanju številke in ključne besede [termostat], program dekodira sporočilo oziroma ukaz, ki ga je dobil. Večino podprtih ukazov komunikacijski demon le posreduje termostatu. Na komunikacijski del se nanašajo le naslednji ukazi:

add_number X Ukaz `add_number X` doda številko `X` v bazo dovoljenih telefonskih števil. Baza telefonskih števil je zapisana v datoteko.

remove_number X Ukaz `remove_number X` iz baze odstrani številko `X`.

valid_number X Ukaz `valid_number` kot odgovor pošlje vse veljavne številke, ki so zapisane v bazi telefonskih števil.

4.1.2 Termostat

Drugi program skrbi za delovanje termostata. V to je vključeno vse od sprejemanje ukazov od komunikacijskega programa, do izvrševanja ukazov in nadzora sistema. V programu uporabljamo knjižnico BCM2835.

Knjižnica BCM2835 nam omogoča uporabo pinov GPIO. Uporaba knjižnice v kodi je zelo enostavna. Na začetku programa jo inicializiramo s funkcijo `bcm2835_init()`. Skozi kodo uporabljamo še nekaj drugih funkcij. S funkcijo `bcm2835_gpio_fsel()` nastavimo določen pin kot vhod ali izhod. Funkcija `bcm2835_gpio_write()` nam omogoča prižiganje ali ugašanje določenega izhoda. Za branje, v katerem stanju, je določen vhod uporabljamo funkcijo `bcm2835_gpio_lev()`. Knjižnica ponuja še ogromno drugih funkcionalnosti, ki jih ne uporabljamo.

Gonilniki za 1-Wire protokol nam omogočajo enostavno branje trenutne temperature, ki jo je izmeril termometer. Gonilniki so narejeni tako, da se izmerjena temperatura zapisuje v datoteko. Če hočemo odčitati temperaturo,

moramo enostavno odpreti datoteko in iz nje prebrati trenutno temperaturo. Osnovna naloga programa je periodično preverjanje temperature, prižiganje in ugašanje centralnega ogrevanja ter branje stanja vhodov. Da se izognemo prekomernemu preklapljanju centralnega ogrevanja, smo uporabili histerezo zanko. Termostat podpira naslednje ukaze:

status

Ukaz status komunikacijskemu demonu sporoči trenutno stanje termostata. To vključuje trenutno temperaturo, ciljno temperaturo, pin na katerem je priključen rele, stanje releja (ali je centralno ogrevanje prižgano ali ne) ter vsa stanja vhodov in izhodov. To potem komunikacijski demon naprej posreduje preko sporočila SMS. Pri večini ukazov je odgovor termostata trenutno stanje.

temperature X

Ukaz temperature X nastavi temperaturo na X °C. Termostat odgovori s sporočilom o trenutnem stanju termostata.

output X on/off

Ukaz vklopi ali izklopi izhod številka X. Ta pin mora biti pred tem nastavljen kot izhod. Termostat odgovori s sporočilom o trenutnem stanju termostata.

new output/input/none X

Ukaz nastavi pin številka X kot vhod, izhod ali kot neuporabljen. Termostat odgovori s sporočilom o trenutnem stanju termostata.

relay X

Nastavi pin X kot pin, na katerega je priklopljen rele, ki vklopi oziroma izklopi centralno ogrevanje. Termostat odgovori s sporočilom o trenutnem stanju termostata.

configuration show/save

Ukaz show pošlje odgovor z opisom trenutne konfiguracije. Save pa to konfiguracijo shrani v lokalno datoteko, iz katere ob zagonu vzpostavi začetno stanje.

4.1.3 Periodično beleženje

Za boljši nadzor in pregledovanje, kaj vse se dogaja na termostatu, imamo periodično beleženje dogodkov. Za večjo preglednost smo naredili več različnih nivojev beleženja. Če bi vse beležili v eno datoteko, bi bilo pregledovanje le te zelo zamudno.

Beleženje dogodkov

Beleženje dogodkov zabeleži vse dogodke, ki se zgodijo na termostatu. Beležimo vse ukaze, ki jih dobi demon termostat. Beležijo se tudi spremembe prižiganja in ugašanja centralnega ogrevanja. Ob vsakem zagonu se v datoteko zapiše datum in čas zagona termostata. Za vsak zabeležen dogodek zabeležimo tudi uro in datum. Primer zapisa beleženja dogodkov vidimo na sliki 4.1.

```
*****
*****      Thermostat start - 07/09/14 18:38:18      *****
*****
-----
07/09/14 18:38:25
Changed target temperature from 25.0°C to 20.0°C

-----
07/09/14 18:38:28
Changed target temperature from 20.0°C to 27.0°C

-----
07/09/14 18:38:28
Turning heater on
Current temperature: 23.5°C
Target temperature: 27.0°C
```

Slika 4.1: Prikaz beleženja dogodkov

Beleženje napak

Beleženje napak smo ločili od beleženja dogodkov, zaradi preglednosti. Ob združenem beleženju napak in dogodkov bi bil pregled, če termostat deluje brez napak, zelo časovno zamuden, saj bi morali pregledati vse dogodke. Pri ločenem beleženju pa enostavno pogledamo eno datoteko in vidimo zbrane vse napake. Tudi pri beleženju napak imamo zapisan čas napake. Na sliki 4.2 vidimo primer zapisa napake.

```
*****
*****      Thermostat start - 28/08/14 17:09:53      *****
*****
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ERROR 28/08/14 17:09:53
Could not read temperature sensor
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ERROR 28/08/14 17:09:59
Could not read temperature sensor
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Slika 4.2: Prikaz beleženja napak

Beleženje sporočil SMS

Za natančen pregled dogajanja si beležimo tudi prejeta sporočila SMS. Beležimo samo sporočila, ki se nanašajo na termostat (se pravi so poslana iz veljavne številke in se začnejo s ključno besedo [termostat]). Beležimo vsa sporočila, ki jih komunikacijski demon pošlje kot odgovore. Obstaja možnost, da pošiljanje sporočila ne uspe (na primer zaradi izgube signala), v tem primeru se zabeleži, da sporočilo ni bilo poslano. Primer beleženja sporočil SMS sporočil vidimo na sliki 4.3.

```

*****
***** Thermostat start - 07/09/14 18:38:18 *****
*****
-----
      Recieved sms
Time:      07/09/14 18:38:25
Number:    041941624
Message:    [termostat] temperature 20
-----
      Sent sms
Time:      07/09/14 18:38:25
Number:    041941624
Message:    c_t 23.5
t_t 20.0
r_p 18
r_s 0
ins
9 0
outs
17 0
22 0

```

Slika 4.3: Prikaz beleženja sporočil SMS

Beleženje temperature

Periodično beležimo tudi temperaturo. Vsak dan se izvedejo štirje zapisi trenutne temperature, ciljne temperature in stanje releja. Te podatke zapišemo vsakih šest ur. Zabeležijo se tudi vse spremembe releja. Beleženje se izvaja predvsem z idejo, da lahko iz teh podatkov potem naredimo graf iz katerega vidimo ciljno, trenutno temperaturo in stanje releja skozi določeno obdobje. Primer beleženja temperature vidimo na sliki 4.4.

```

*****
***** Thermostat start - 07/09/14 18:38:18 *****
*****
07/09/14 18:38:25 23.5°C 20.0°C 0
07/09/14 18:38:28 23.5°C 20.0°C 0
07/09/14 18:38:28 23.5°C 27.0°C 0
07/09/14 18:38:28 23.5°C 27.0°C 1
07/09/14 18:38:35 24.1°C 25.0°C 1
07/09/14 18:39:11 27.1°C 25.0°C 0
07/09/14 18:39:28 25.8°C 25.0°C 0
07/09/14 18:40:28 23.9°C 25.0°C 0
07/09/14 18:41:28 23.3°C 25.0°C 0
07/09/14 18:42:28 23.2°C 25.0°C 0
07/09/14 18:43:15 22.9°C 25.0°C 1

```

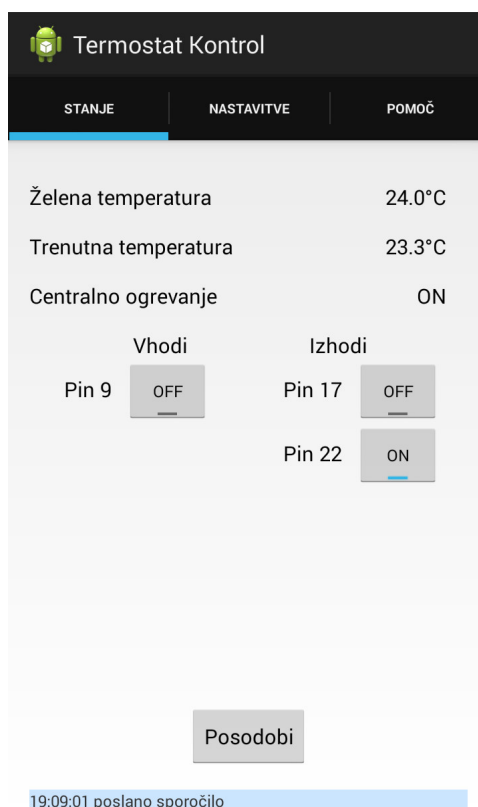
Slika 4.4: Prikaz beleženja temperature

4.2 Mobilna aplikacija

Termostat je mogoče nadzorovati s pomočjo sporočil SMS. Vendar je to uporabniku dokaj neprijazen način upravljanja, saj bi si uporabnik moral vse ukaze zapomniti. Odločili smo se za izdelavo aplikacije, ki nam to olajša. Aplikacija glede na naše zahteve izdelava sporočila SMS in jih pošilja termostatu. Aplikacijo smo izdelali na mobilni platformi Android. Aplikacija ima tri zavihke. Na prvem zavihku imamo prikazano trenutno stanje, drug zavihek služi za nastavitve, tretji pa je namenjen pomoči. Za premikanje med zavihki lahko uporabimo geste potegnite levo oziroma potegnite desno (angl. Swipe left oziroma Swipe right), lahko pa uporabimo zavihke na vrhu zaslona.

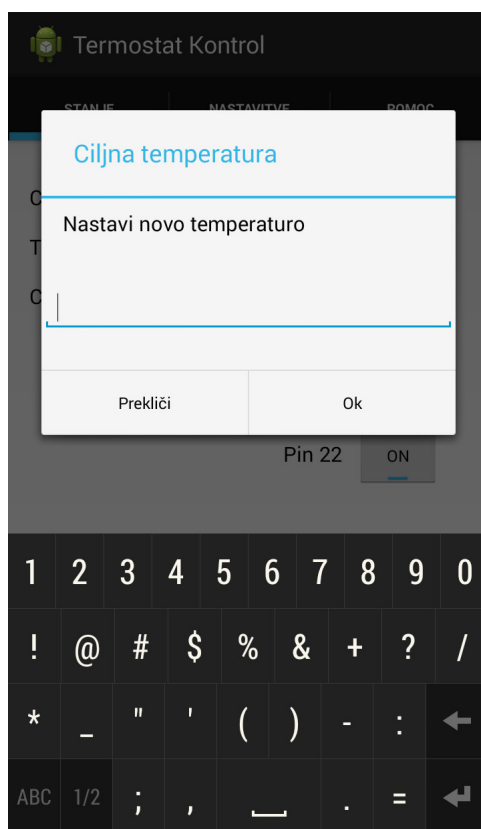
4.2.1 Stanje

Na prvem zavihku imamo prikazano trenutno stanje. Postavitve lahko vidimo na sliki 4.5.



Slika 4.5: Prikaz postavitve zavihka stanje

Na zaslonu imamo izpisano želeno temperaturo, trenutno temperaturo, ali je centralno ogrevanje prižgano ali ne ter stanje vseh vhodov in izhodov. Na dnu zaslona imamo gumb Posodobi ter statusno vrstico. Ob pritisku na ciljno temperaturo se nam odpre okence, ki ga vidimo na sliki 4.6, v katerega lahko vpišemo novo ciljno temperaturo.



Slika 4.6: Prikaz okenca za spremembo temperature

Ob vsakem izhodu imamo preklopni gumb. Z njim lahko vklapljammo in izklapljammo izhod. Lahko bi se odločili, da aplikacija pošlje sporočilo SMS ob vsaki spremembi, vendar smo se rajši odločili, da aplikacija pošlje sporočilo samo ob pritisku na gumb Posodobi. Tak pristop smo uporabili zaradi zmanjšanja stroškov. Omogoča nam, da uporabnik nastavi vse želene spremembe in z gumbom Posodobi vse spremembe pošljemo v enem sporočilu SMS. Če bi na primer uporabnik spremenil ciljno temperaturo in dva izhoda, bi v prvem primeru poslali tri sporočila SMS, tako pa pošljemo samo enega. Pojavi pa se problem, da ne moremo točno vedeti, kateri podatek je posodobljen in kateri spremenjen iz naše strani. Zato se ob vsaki spremembi spremenjen element obarva v zeleno. Ob posodobitvi pa se ponovno obarva v barvo ozadja. Primer spremenjenega elementa vidimo na sliki 4.7.



Slika 4.7: Prikaz spremenjenega elementa

V statusni vrstici imamo napisan čas, kdaj je bil program nazadnje posodobljen.

4.2.2 Nastavitve

Drugi zavihek nam omogoča nastavljanje termostata. Njegovo postavitve vidimo na sliki 4.8.



Slika 4.8: Prikaz postavitve zavihka nastavitve

Prva zadeva, ki jo lahko nastavimo je številka termostata. To je telefonska številka, na katero bo aplikacija pošljala sporočila SMS. Naslednja nastavitve so veljavne telefonske številke. To so vse telefonske številke, iz katerih bo termostat sprejemal ukaze. Pri veljavnih telefonskih številkah imamo dva gumba, gumb Posodobi in gumb Dodaj številko. Gumb Posodobi od termostata zahteva seznam vseh veljavnih številke. Posodabljanje seznama številke smo realizirali ločeno od ostalega posodabljanja. Razlog za to je, da se seznam številke načeloma ne spreminja pogosto in bi bilo potratno, če bi ga vedno pošljali skupaj s stanjem termostata. Drugi gumb je gumb Dodaj številko. Ob pritisku na gumb se odpre okence, v katerega zapišemo telefonsko številko, ki jo želimo dodati. Naslednji med nastavitvami je pin, na katerega je povezan rele, s katerim kontroliramo centralno ogrevanje. Po-

dobno kot pri nastavitvi temperature s klikom na številko odpremo novo okence, v katerega vpišemo novo številko pina za rele. Sledijo še vhodi in izhodi. Pri obeh imamo listo trenutno določenih pinov za vhod oziroma izhod. Imamo tudi dva gumba Dodaj, s katerima dodamo nov vhod oziroma izhod. Na dnu zavihka vidimo že znan gumb Posodobi in statusno vrstico. Tudi na tem zavihku se pri vsaki spremembi spremenjen element pobarva z zeleno.

4.2.3 Pomoč

Zadnji zavihhek služi za pomoč uporabnikom. Na njem imamo zapisana navodila za uporabo programa. Prikaz zavihka si lahko ogledamo na sliki 4.9.



Slika 4.9: Prikaz postavitve zavihka pomoč

Poglavje 5

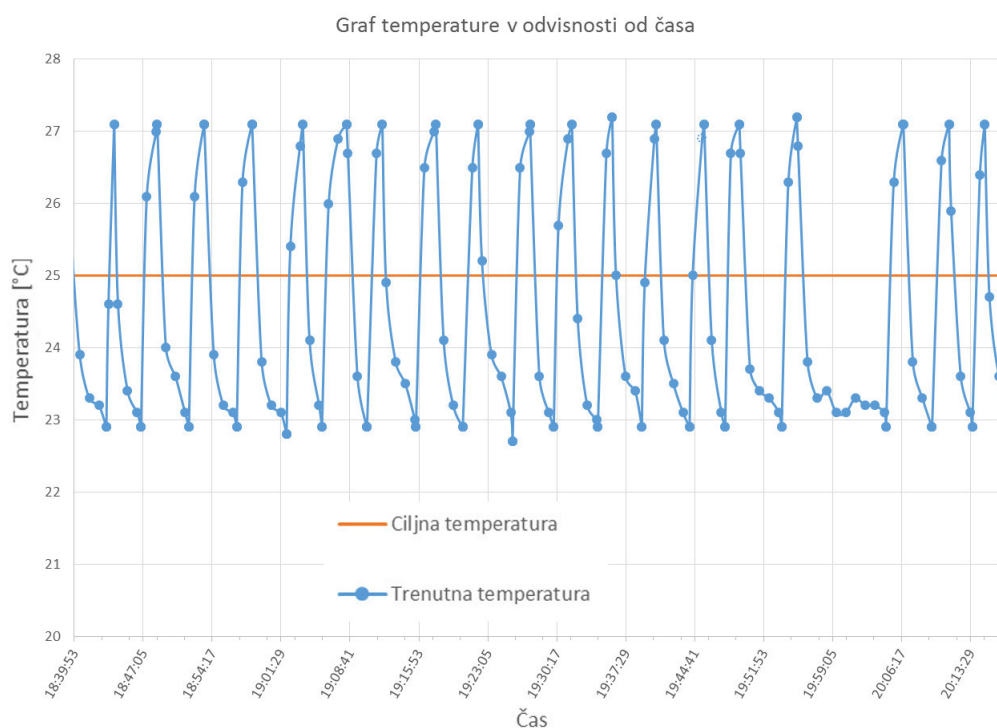
Rezultati

Za testiranje končanega izdelka smo izdelali testno platformo. Uporabili smo naše testno vezje, s katerim smo lahko preizkusili vhode in izhode. Za preverjanje delovanja algoritma in spreminjanje temperature smo potrebovali grelno telo. Odločili smo se, da ga bomo izdelali kar sami. Uporabili smo grelno žico, ki smo jo ovili okoli lepenke, kot vidimo na sliki 5.1. To smo preko releja priklopili na napetost petih voltov. Ker je grelno telo majhno, smo ga postavili blizu termometra za največji učinek. S tem smo simulirali prižiganje centralnega ogrevanja in posledično ogrevanja prostora.



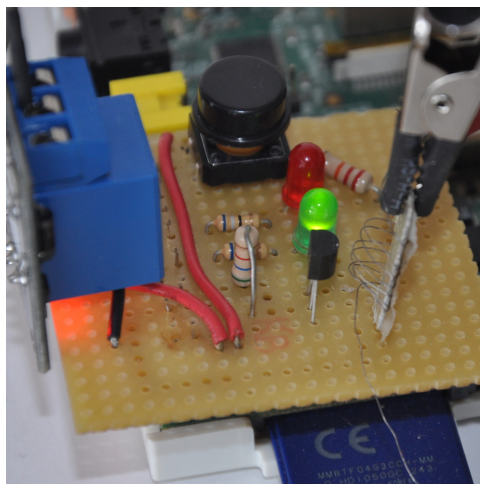
Slika 5.1: Grelno telo za testiranje

Testirali smo pri želeni temperaturi 25°C . Testirali smo na dobro uro in pol, pri tem pa smo zbrali nekaj več kot 130 meritev. Na grafu 5.1 lepo vidimo prikaz delovanja histerezne zanke. Ogrevanje termostat vklopi pri temperaturi nižji od 23°C , izklopi pa ga, ko je temperatura višja od 27°C . Iz grafa lahko tudi opazimo da je ohlajanje počasnejše od ogrevanja.



Graf 5.1: Graf trenutne temperature in ciljne temperature v odvisnosti s časom

Na sliki 5.3 vidimo prikazano sliko aplikacije, na sliki 5.2 pa testno vezje med testiranjem. Prikazan je prižgan izhod, ciljna temperatura nastavljena na 27°C in trenutna temperatura $23,6^{\circ}\text{C}$. Zato je rele prižgan in grelna telo ogreva okolico termometra.



Slika 5.2: Prikaz testnega vezja med delovanjem



Slika 5.3: Prikaz aplikacije med delovanjem

Poglavje 6

Zaključek

Cilj diplomske naloge je bil izdelati termostat z možnostjo oddaljenega krmljenja preko sporočil SMS. Zaradi večje namembnosti smo termostatu dodali še možnost nadzorovanja digitalnih vhodov in izhodov. Naš izdelek tako omogoča oddaljeno spreminjanje zelene temperature in nadzor digitalnih vhodov ter izhodov. Oddaljen nadzor je realiziran s pomočjo sporočil SMS. Za bolj prijetno uporabniško izkušnjo smo naredili mobilno aplikacijo na operacijskem sistemu Android. S pomočjo aplikacije lahko naše zahteve enostavno vnesemo v uporabniški vmesnik aplikacije, ki nato namesto nas izdelava sporočilo SMS in ga pošlje termostatu. Cena vseh komponent našega termostata ne presegajo 100 evrov. Če to primerjamo s ceno obstoječih rešitev, ki se giblje od 240 evrov naprej, vidimo, da je naša rešitev cenovno zelo ugodna. S končnim izdelkom smo zadovoljni, saj omogoča vse, kar smo si zamislili na začetku diplome. S pomočjo mobilne aplikacije nam omogoča boljšo uporabniško izkušnjo oddaljenega nadzora, kot obstoječe rešitve. Nadzor digitalnih vhodov in izhodov pa razširita njegovo uporabnost. Vendar pa kot vsaka stvar ima tudi naš termostat možnosti za izboljšave.

Idej, kako naš projekt izboljšati oziroma nadgraditi, je veliko. Zelo uporabno bi bilo, če bi na računalnik Raspberry Pi priključili 16x2 segmentni zaslon, na katerega bi izpisovali pomembne podatke. S tem bi omogočili nadzorovanje termostata tudi na lokaciji. Zanimiva opcija bi bila tudi programiranje

brezžičnih temperaturnih senzorjev, saj bi tako lahko dobili malo več podatkov o trenutnih temperaturah po različnih prostorih zgradbe. Kar se tiče aplikacije Android bi jo lahko izboljšali tako, da bi lahko vhode in izhode preimenovali. Tako bi imeli namesto pin 18 napisano Luč pri garaži. S takimi imeni bi izboljšali uporabniško izkušnjo.

Literatura

- [1] C Andrews. Easy as Pi [Raspberry Pi]. *Engineering & Technology*, 8(3):34–37, 2013.
- [2] Audon. BPT322-GST Wireless Thermostat Programmer - text message heating control, 2014. [Online; accessed 1-September-2014].
- [3] Siop Elektronika. Mars 2, 2014. [Online; accessed 1-September-2014].
- [4] Stephen Bo Furber. *ARM system-on-chip architecture*. pearson Education, 2000.
- [5] M. Hegger, A. Reichel, K. Schultz, J. Hartwig, and M. Keller. *Heat - Cool: Energy Concepts, Principles, Installations*. Scale (Basel, Switzerland). Birkhäuser, 2012.
- [6] B. Horan. A Simple Temperature Sensor. In *Practical Raspberry Pi*, Technology in action series. Apress, 2013.
- [7] W.S. Levine. *Control System Applications*. Taylor & Francis, 1999.
- [8] Picture of Raspberry Pi model B, 2014. [Online; accessed 22-August-2014].
- [9] Picture of relay board, 2014. [Online; accessed 22-August-2014].
- [10] Picture of thermometer with 1-wire protocol, 2014. [Online; accessed 22-August-2014].

- [11] B. Smith. *Linux Appliance Design: A Hands-on Guide to Building Linux Appliances*. No Starch Press Series. No Starch Press, 2007.
- [12] Wikipedia. Hysteresis — wikipedia, the free encyclopedia, 2014. [Online; accessed 22-August-2014].